

1.RStudio개발환경

다른 언어와 비교하여 R을 지원하는 개발환경/도구는 그리 많은 편은 아니다. 그중에서 RStudio 는 다른 R개발툴과 비교하여 멋진 그래픽 인터페이스를 제공하고 편리한 개발을 지원하기 때문에 R을 사용한다면 반드시 설치해서 사용할 개발도구이다. RStudio는 R을 이용한 프로그래밍을 더욱 생산적으로 만들고 직관적으로 사용하기 쉽도록 여러 가지 기능을 갖춘 개발환경이다.

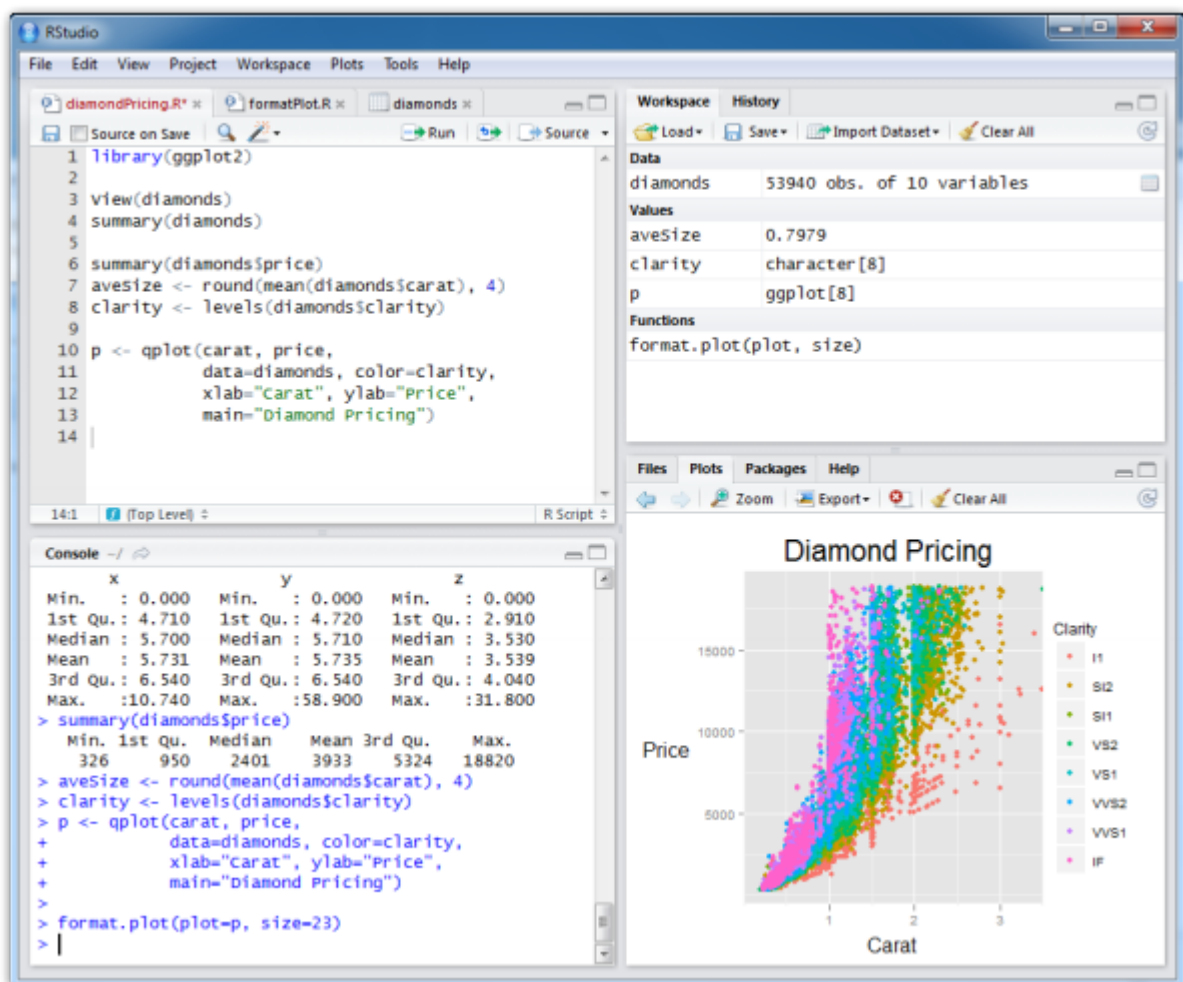
기능상 RStudio는 다음과 기능을 가진다.

Plot캐싱-그래픽 출력에 대한 정보를 가지고 있어 이전에 작성한 그래픽도 재 실행하지 않고 볼 수 있다.

코드완성기능 - 함수, 개체, 변수목록을 제공하고 사용자가 작성한 함수도 지원한다

1.1 RStudio 화면구성

RStudio는 보통 [Source], [Console], [Workspace, History], [Files, Plots, Packages, Help] 등 네 개의 화면(Pane)으로 구성된다(Pane의 구성은 옵션설정에 따라 달리 구성할 수 있다. 여기에서 소개한 Pane은 옵션을 변경하지 않는 기본적인 구성에 따른 것이다)



▣ [Source] Pane

[Source]는 프로그램을 작성하는 에디터이다. 여기에서 R 프로그램 스크립트를 작성할 수 있으며, 데이터프레임과 같은 데이터를 보여주기도 한다.

▣ [Console] Pane

[Console]은 R의 명령어를 입력하여 결과를 바로 바로 볼 수 있는 공간이다. 간단한 연산 결과를 알고 싶거나 함수를 사용하는 경우에 유용하다. 또한 R스크립트의 실행결과를 출력하는 장소이기도 하다.

▣ [Workspace, History, VCS] Pane

[Workspace, History, VCS]는 R 스크립트에서 사용하는 변수와 값 그리고 스크립트에 있는 함수목록을 보여주는 [Workspace]탭, [Console] Pane에서 입력했던 명령어 내역 등을 보관하는 [History]탭 등으로 구성된다.

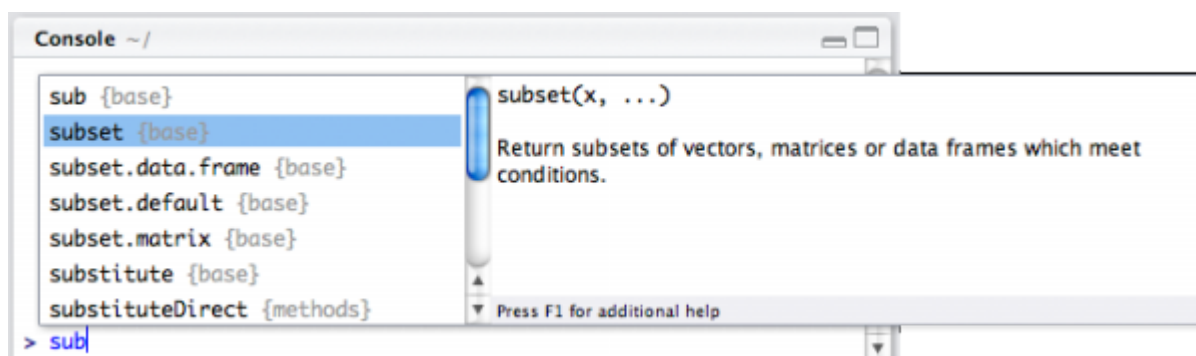
[Files, Plots, Packages, Help]

2.Console에서 작업하기

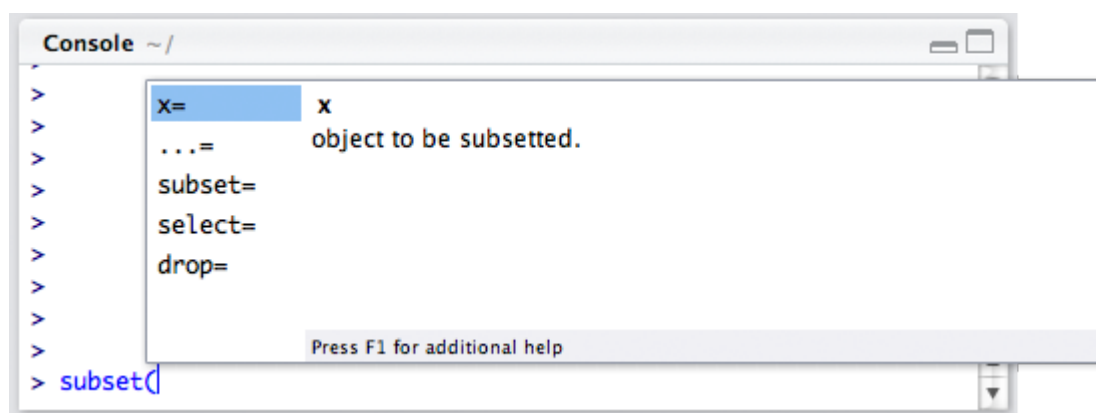
▣ 코드 자동완성

[TAB]키를 사용하여 코드를 자동완성하는 기능을 제공한다. 가령 R개체나 함수이름의 일부를 입력해두고 [TAB]키를 입력하면 RStudio 는 자동으로 그 이름으로 시작하는 개체나 함수이름을 제시한다.

가령 sub만 입력해두고 [TAB]을 입력하면 subset, subset.data.frame, subset.default 등등을 제시한다.

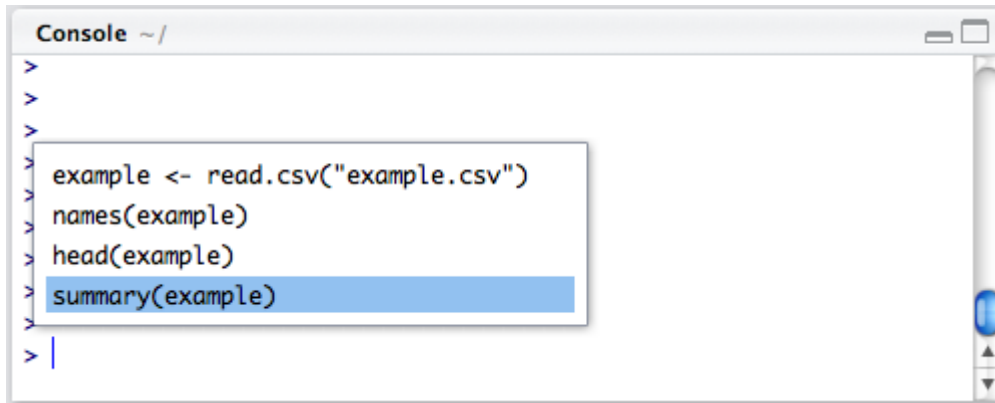


함수를 사용하는 경우 자주 사용하지 않거나 새로 익히는 경우라면 함수에 전달할 인수가 기억나지 않을 수 있다. 이 경우 함수이름과 괄호까지만 입력해두고 [TAB]키를 입력하면 함수사용에 필요한 인수에 대한 정보를 얻을 수 있다.



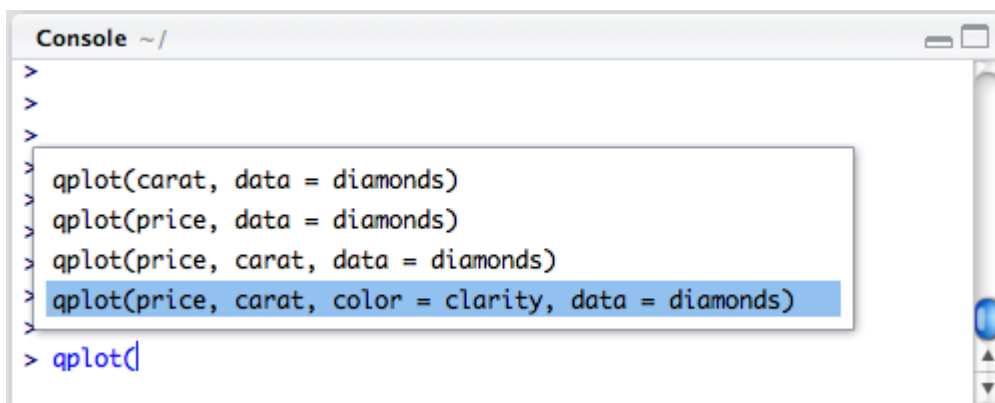
▣ 이전 입력 불러오기

R의 기본 개발환경인 [RGui]와 마찬가지로 상하 방향 키를 사용하여 이전에 사용한 명령을 불러올 수 있으며 [CTRL]+[UP] 단축 키를 사용하면 이전 명령을 보여주는 팝업메뉴를 볼 수 있다.



```
Console ~/>
>
>
> example <- read.csv("example.csv")
> names(example)
> head(example)
> summary(example)
> |
```

[CTRL]+[UP] 단축 키는 이전 명령과 유사한 명령목록을 보여주기도 한다. 아래의 그림은 여러 번의 `qplot` 함수 사용내역을 보여준다.



```
Console ~/>
>
>
> qplot(carat, data = diamonds)
> qplot(price, data = diamonds)
> qplot(price, carat, data = diamonds)
> qplot(price, carat, color = clarity, data = diamonds)
> qplot(|
```

[Console] pane의 제목표시줄에는 몇 가지 기능을 가지고 있다. 우선 'Console'이라는 표시 뒤 회색 글씨는 작업하고 있는 작업 폴더를 가리킨다. 또한 [STOP]버튼은 현재 실행중인 R스크립트를 강제로 중단하는 역할을 한다. 그외 나머지 두 개의 버튼은 [Console] pane의 크기를 늘리거나 줄이는 역할을 한다(다른 pane의 것도 같은 기능이다.)



▣ 기타 단축 키

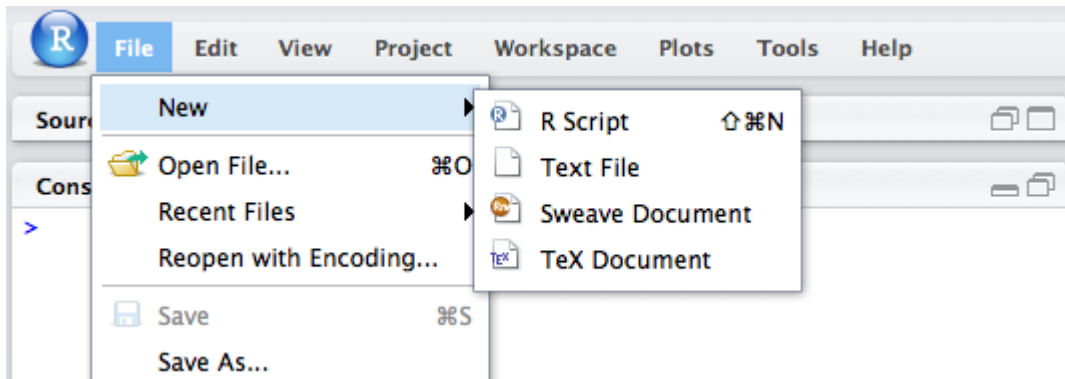
- [CTRL]+[1] : [Source] 로 이동
- [CTRL]+[2] : [Console] 로 이동
- [CTRL]+[L] : [Console] 내용 지움
- [Esc] : R 스크립트 실행 중단

3.편집 및 코드 실행하기

RStudio의 에디터는 구문강조, 코드완성, 여러 파일 편집, 찾기/바꾸기 등 작업의 효율성을 높여주는 여러 기능을 가지고 있다. 이러한 기능은 R소스코드외에 Sweave 파일, TeX파일에서도 지원한다.

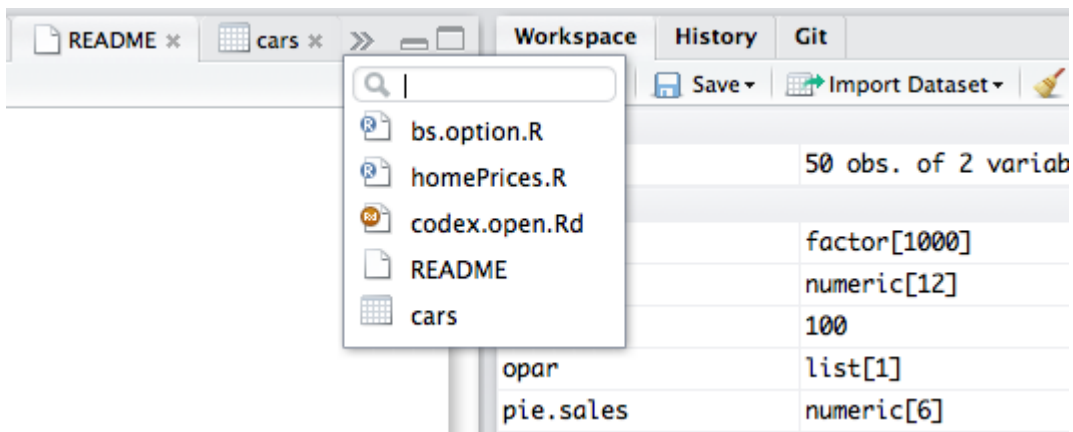
3.1 파일

새로운 R소스, Sweave 파일, TeX파일을 만들려면 [File]-[New]를 클릭한다.



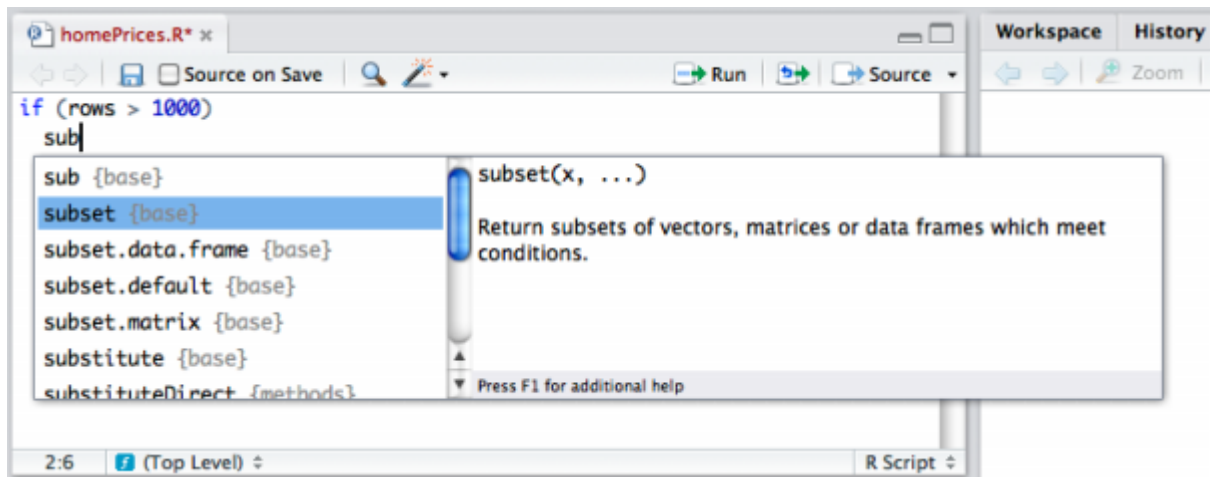
저장해둔 파일을 여는 경우 [File]-[Open File]을 클릭한다. 또는 최근에 열어 본 목록중 파일을 선택한다면 [File]-[Recent Files]를 이용할 수 있다.

RStudio는 동시에 여러 개의 파일을 열어 편집할 수 있는데, 에디터 상단의 탭을 이용하여 파일과 파일 사이를 이동하여 작업할 수 있다. 파일을 많이 열어 둔 경우 에디터 상단 오른쪽 끝의 >> 아이콘을 이용하여 원하는 파일로 이동할 수 있다. 또는 [View]-[Switch to Tab]을 사용해도 된다.



3.2 코드완성

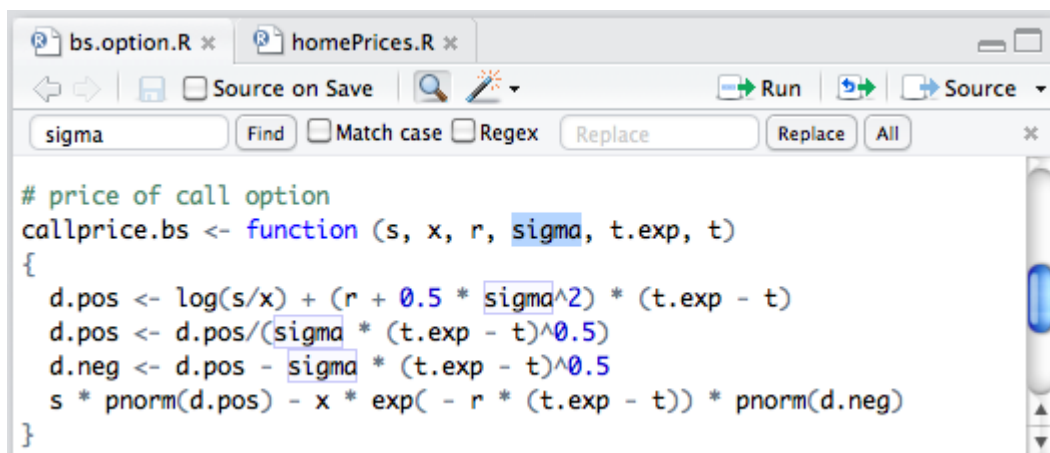
RStudio는 [Tab]키를 이용하여 작성중인 코드를 자동으로 완성하는 기능을 제공한다. 가령 'sub'라고 입력하고 [Tab]키를 누르면 'sub'로 시작하는 개체나 함수를 보여준다.



코드완성기능은 에디터외에 콘솔창에서도 사용가능하다.

3.3 찾기/바꾸기

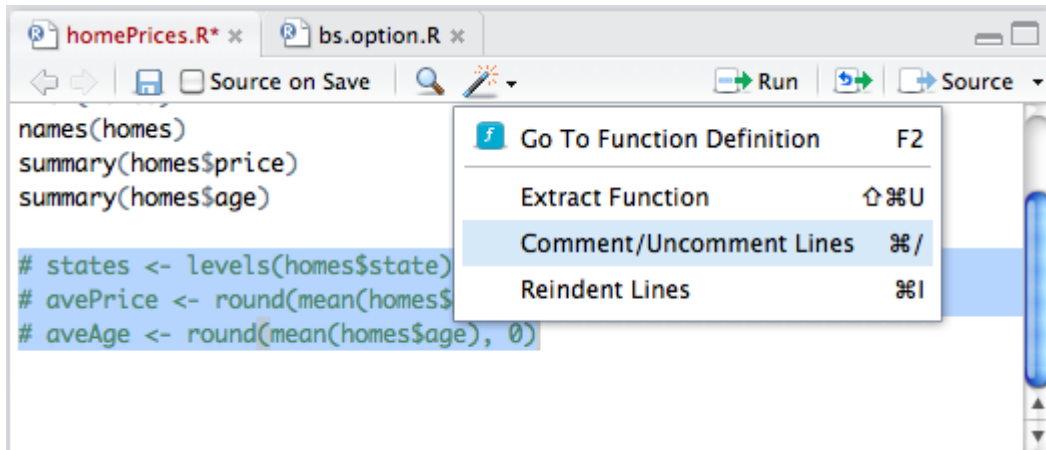
RStudio는 소스파일내에서 찾기/바꾸기를 제공한다.



단축키 [Ctrl]+[F] 또는 [Edit]-[Find and Replace]를 사용하면 된다.

3.4 주석달기/해제

프로그램에 대한 설명을 적거나 원하는 코드를 주석을 처리하여 실행을 임시로 막는 경우 주석달기/해제 기능을 이용할 수 있다. 주석을 달거나 해제하려면 [Edit]-[Comment/Uncomment Lines] 또는 단축키 [Ctrl]+[I]를 이용한다.



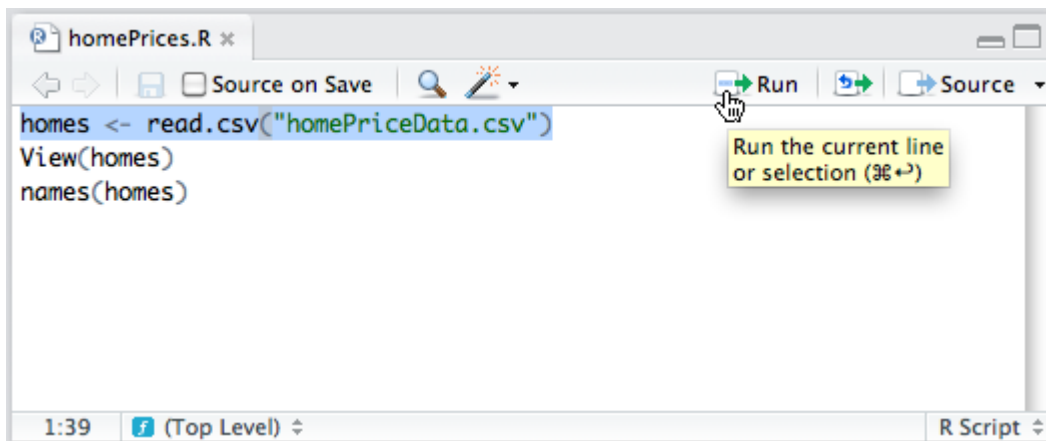
3.5 들여쓰기

RStudio는 코드를 붙여 넣거나 코드 블록의 시작을 알리는 '{'에서 [Enter]를 눌러 줄을 바꾸는 경우 자동으로 들여쓰기를 해준다. [Edit]-[Reindent Lines] 또는 단축 키 [Ctrl]+[I]를 사용한다.

3.6 코드 실행

RStudio에서는 여섯 가지의 종류의 코드 실행방법이 있다.

현재 커서가 위치해 있거나 선택한 라인만 실행하는 경우 [Edit]-[Run Code]-[Run Line(s)] 또는 단축 키 [Ctrl]+[Enter] 또는 도구모음의 [Run]버튼을 클릭한다.



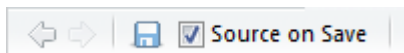
코드 실행이 끝나면 커서는 실행이 끝난 이후의 지점으로 이동하여 [Ctrl]+[Enter]를 다시 사용하여 바로 한 줄씩 실행할 수 있게 배려해준다. 그러나 앞서 직전에 실행한 코드를 다시 실행하고자 하는 경우 [Edit]-[Run Code]-[Re-Run Previous Region] 또는 단축 키 [Ctrl]+[Shift]+[P]를 사용한다.

| | |
|----------------------------|--------------|
| Run Line(s) | Ctrl+Enter |
| Run All | Ctrl+Shift+R |
| Run From Beginning To Line | Ctrl+Shift+B |
| Run From Line to End | Ctrl+Shift+E |
| Run Function Definition | Ctrl+Shift+F |
| Re-Run Previous Region | Ctrl+Shift+P |

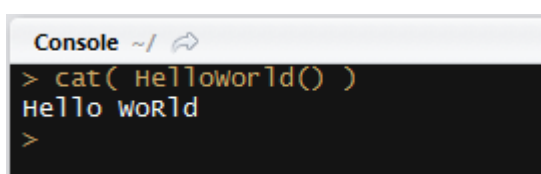
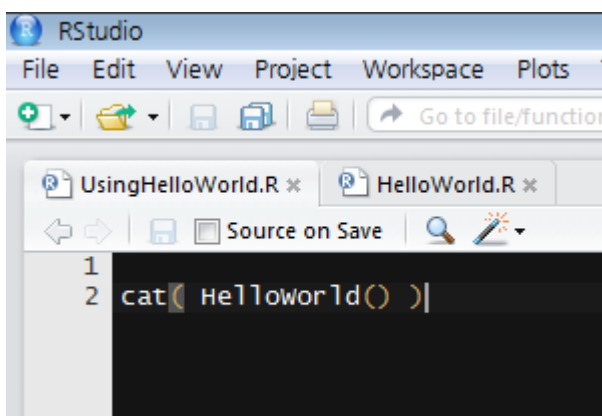
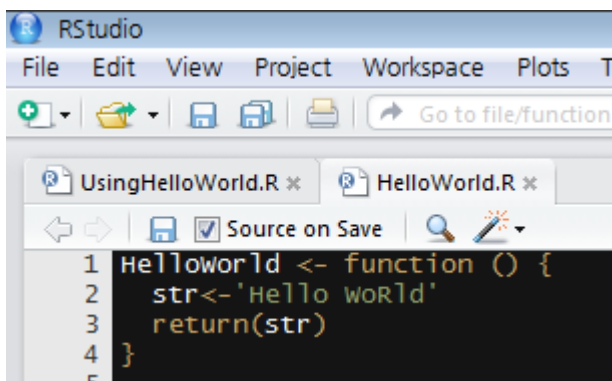
- [Edit]-[Run Code]-[Run Line(s)] / [Ctrl]+[Enter] – 현재 커서가 위치한 행 또는 선택한 행
- [Edit]-[Run Code]-[Run All] / [Ctrl]+[Shift]+[R] – 전체 코드
- [Edit]-[Run Code]-[Run From Beginning To Line] / [Ctrl]+[Shift]+[B] – 첫 행부터 현재 커서가 위치한 행
- [Edit]-[Run Code]-[Run Line To End] / [Ctrl]+[Shift]+[E] – 현재 커서 위치에서 코드의 마지막 행
- [Edit]-[Run Code]-[Run Function Definition] / [Ctrl]+[Shift]+[F] – 현재 커서가 위치한 함수
- [Edit]-[Run Code]-[Re-Run Previous Region] / [Ctrl]+[Shift]+[P] – 직전 실행한 코드

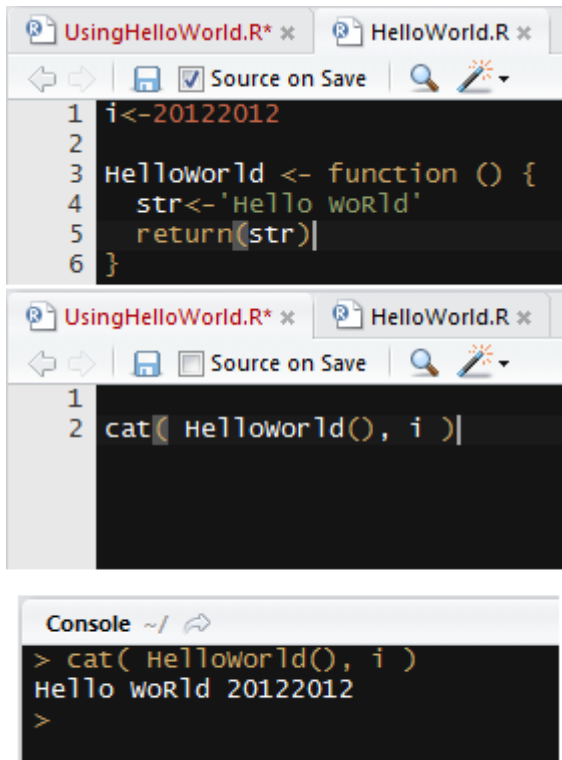
3.7 Source On Save

함수를 재사용하려는 경우 에디터의 도구모음에 위치한 [Source On Save]를 체크해둔다. 이렇게 체크해두면 함수를 수정하여 저장할 때마다 전역환경(global environment)에 보관된다.



이렇게 전역환경에 저장된 함수는 따로 함수가 담긴 파일을 열지 않아도 다른 파일에서 사용할 수 있다. 물론 함수외에 변수도 이런 식으로 사용할 수 있다.





```
1 i<-20122012
2
3 HelloWorld <- function () {
4   str<-'Hello world'
5   return(str)
6 }

1
2 cat( HelloWorld(), i )

Console ~/
> cat( HelloWorld(), i )
Hello world 20122012
>
```

3.8 단축 키

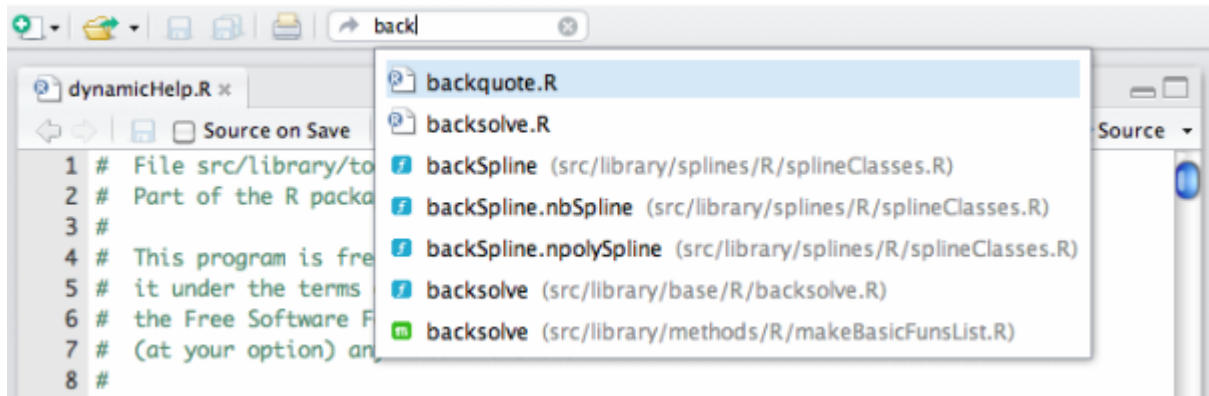
- Ctrl+Shift+N — 새 파일 작성
- Ctrl+O — 파일 열기
- Ctrl+S — 현재 파일 저장
- Ctrl+1 — 소스 에디터로 이동
- Ctrl+2 — 콘솔로 이동

4.코드 편집하기

RStudio는 R의 소스코드를 찾거나 이동하기 위한 편리한 기능을 제공하고 있다.

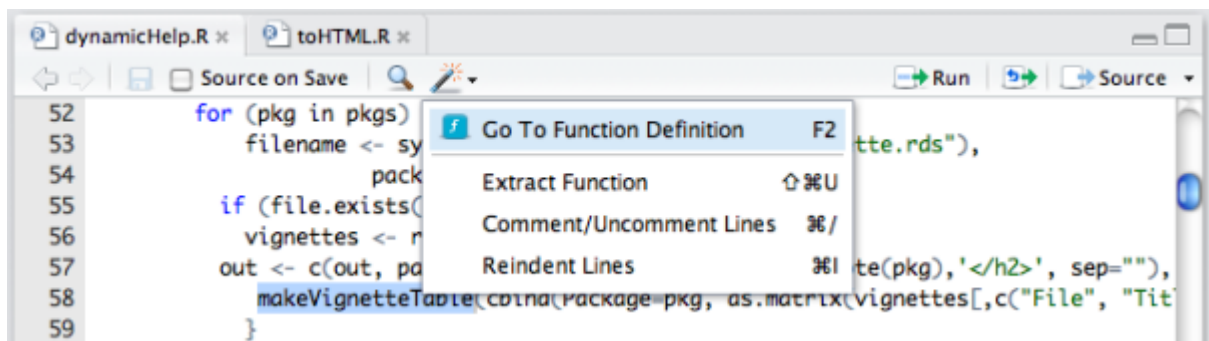
4.1 Go to File/Function

편집하려는 소스파일이나 함수의 이름을 알고 있다면 RStudio 도구모음의 [Go to File/Function]를 이용하여 재빨리 이동할 수 있다. 단축 키는 [Ctrl]+[.] 이다.



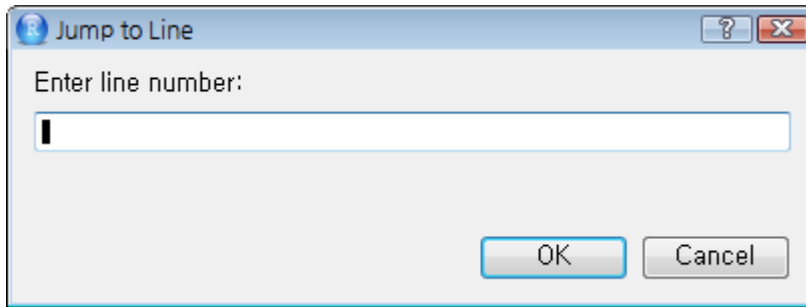
4.2 Go to Function Definition

RStudio는 함수의 정의를 이동할 수 있다. 함수이름에 커서를 위치시키고, 편집기의 도구모음에서 [Code Tools]-[Go to Function Definition]를 클릭하면 함수를 정의한 곳으로 이동할 수 있다. 단축 키는 [F2]이다. 또 다른 방법은 함수이름에 커서를 위치시키고, [Ctrl]키를 누른 상태에서 마우스 왼쪽버튼을 클릭해도 된다.

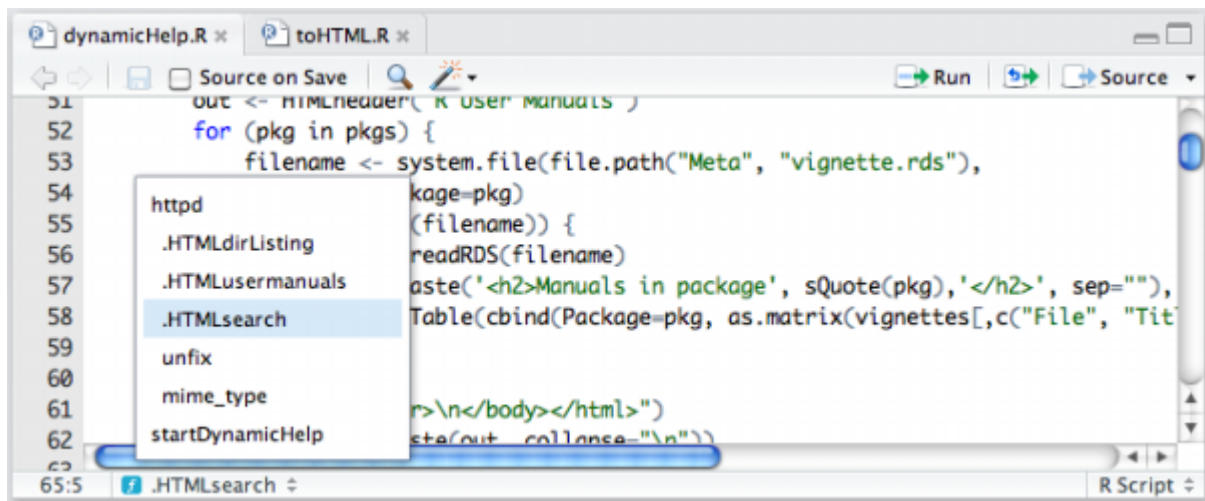


4.3 소스파일내에서 이동하기

소스파일내에서 이동하는 방법에도 앞서 말한 기능외에 몇 가지 방법이 더 있다. [Ctrl]+[G]를 누르면 원하는 행 번호로 이동할 수 있다.



또 편집기 왼쪽 하단의 [f]모양의 버튼을 클릭하면 함수목록이 팝업 창에 표시되고 이중 하나를 클릭하면 해당 함수로 이동한다.



5. 프로젝트 사용하기

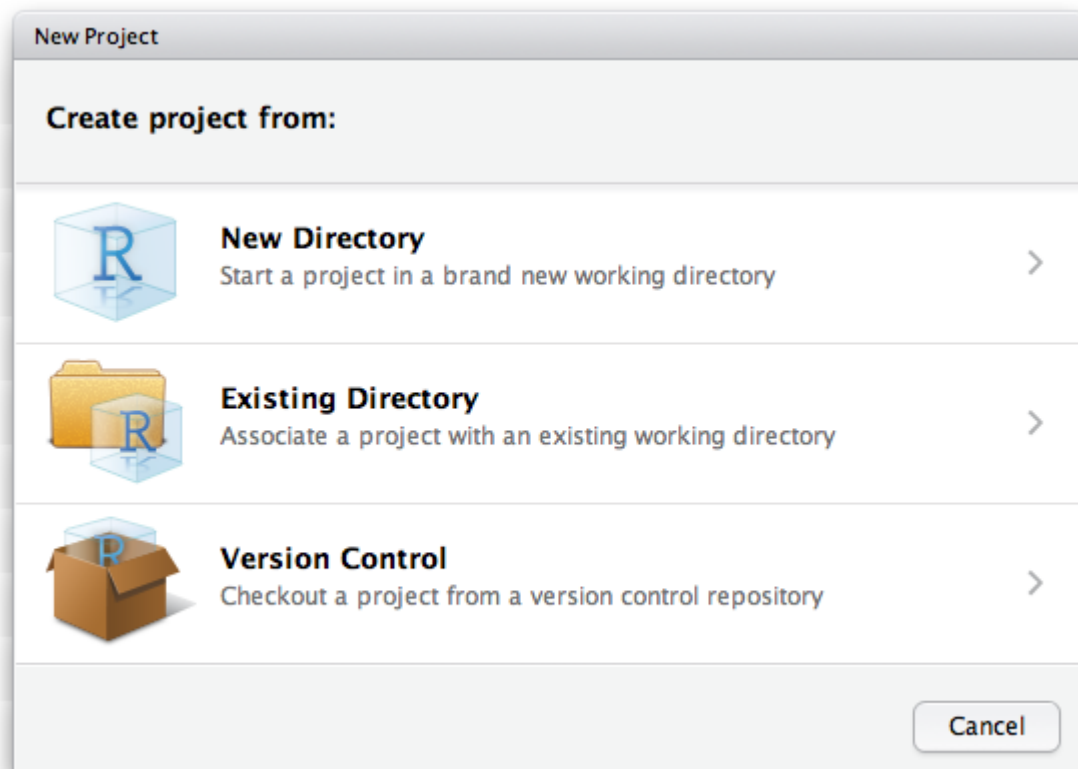
RStudio 의 프로젝트는 하나의 작업을 여러 개로 분리하여 각각 워킹디렉토리, 워크스페이스, 명령어 기록, 소스문서로 관리하는 편리한 기능이다.

5.1 프로젝트 만들기

RStudio 의 프로젝트는 R의 워킹디렉토리를 관리한다. 여러분은 다음과 같은 방법으로 프로젝트로 만들 수 있다.

- ▣ 새로운 폴더
- ▣ R코드와 데이터가 있는 폴더
- ▣ 버전관리저장소(Git 또는 Subversion) 복제

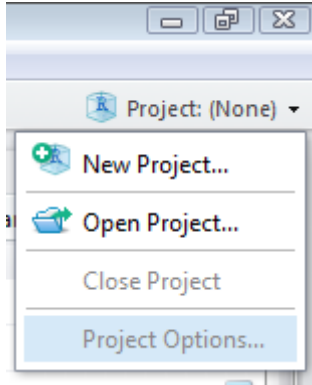
새로운 프로젝트를 만들려면 메뉴에서 [Project]-[New Project]를 클릭하거나 오른쪽 상단의 [Project]를 클릭한다.



새로운 프로젝트가 만들어지면, 프로젝트 디렉토리내 프로젝트파일(*.Rproj)이 만들어진다. 이 파일은 여러 가지 프로젝트 옵션 정보를 가지고 있다. 프로젝트 고유의 임시파일이 생기는 경우 Rproj.user라는 폴더가 생성된다(기본적으로 숨겨져 있다)

5.2 프로젝트 열기

프로젝트를 열려면 메뉴에서 [Project]-[Open Project] 또는 [Project]-[Recent Projects]를 클릭하거나 도구모음의 [Project]를 클릭한다.



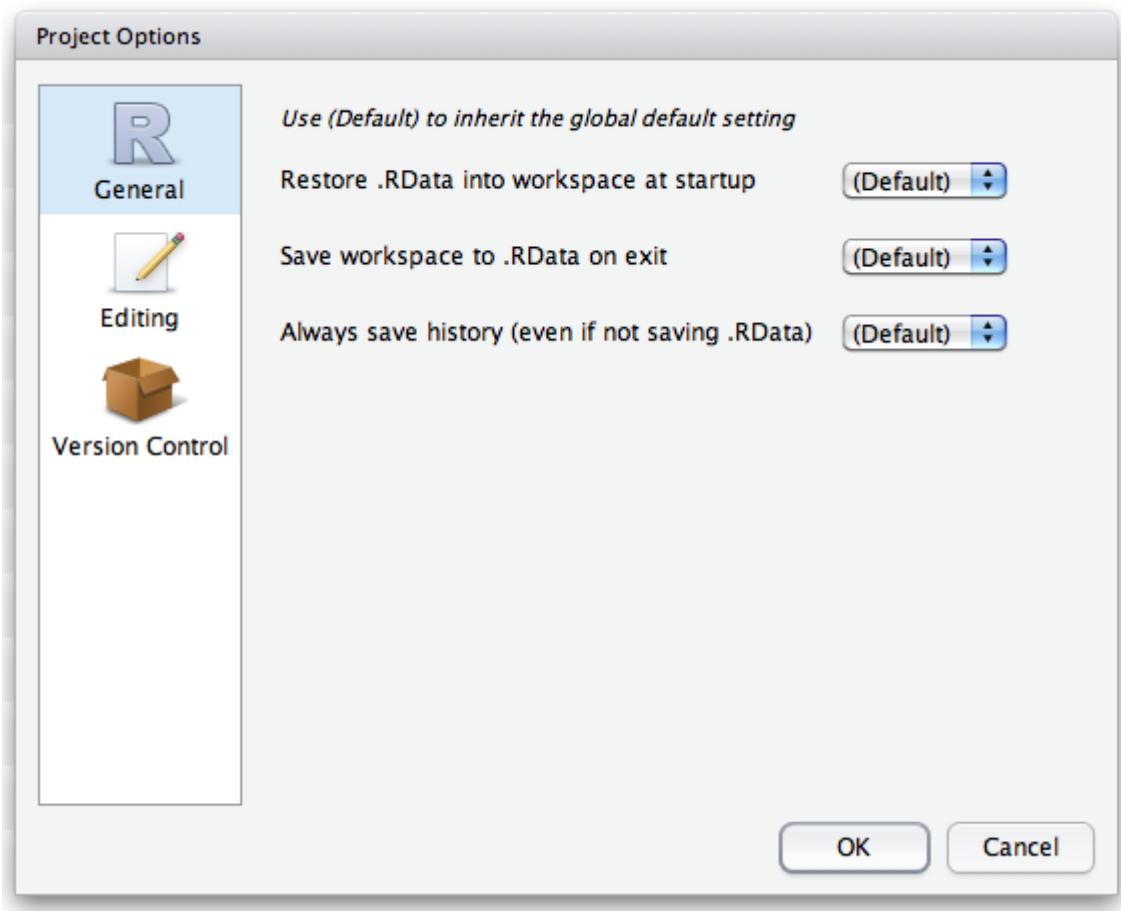
프로젝트파일이 열리면 새로운 R세션이 시작되고 프로젝트 폴더내 .Rprofile파일이 소스화되고 .RData파일이 로딩된다. 또한 .Rhistory파일이 RStudio의 History창에 로드된다. 또한 작업디렉토리는 프로젝트 폴더가 된다. 또한 이전에 저장한 프로젝트의 소스파일이 열리게 된다.

5.3 프로젝트 옵션

몇 가지 프로젝트 옵션이 있는데, 이것은 프로젝트가 로드되거나 종료되거나 저장할 때 RStudio의 작동을 조정할 수 있고 편집과 관련된 설정을 바꿀 수 있다.

[General]

- Restore .RData into workspace at startup — 시작시 .RData 파일을 R 워크스페이스에 로드할 것인지 여부를 정한다. 굉장히 큰 .RData파일이 있다면 로드하지 않도록 하는 것이 좋다.
- Save workspace to .RData on exit — 종료시 .RData파일 저장여부를 선택할 수 있다. 워크스페이스에 변경이 없다면 이 옵션설정에 상관없이 저장여부를 물어 보지 않는다.
- Always save history (even when not saving .RData) — 종료시 .RData파일을 저장하지 않아도 .Rhistory 파일은 항상 저장되도록 설정할 수 있다.



[Editing]

- Index R source files — 프로젝트폴더내 소스파일을 색인화할 것인지를 정한다. 색인을 만들면 코드가동(가령 go to file/function, go to function definition)을 위한 것이다.
- Insert spaces for tab — 코드를 편집하면서 [Tab]키를 누를 때 여러 개의 공백문자로 탭을 만들지 아니면 탭 문자로 할지를 정할 수 있다.
- Text encoding — 소스파일의 텍스트 인코딩방식을 정할 수 있다. 기본 인코딩과 다른 소스파일은 [File]-[Reopen with Encoding]을 통해 인코딩을 수정하여 열 수 있다.

[Version Control]

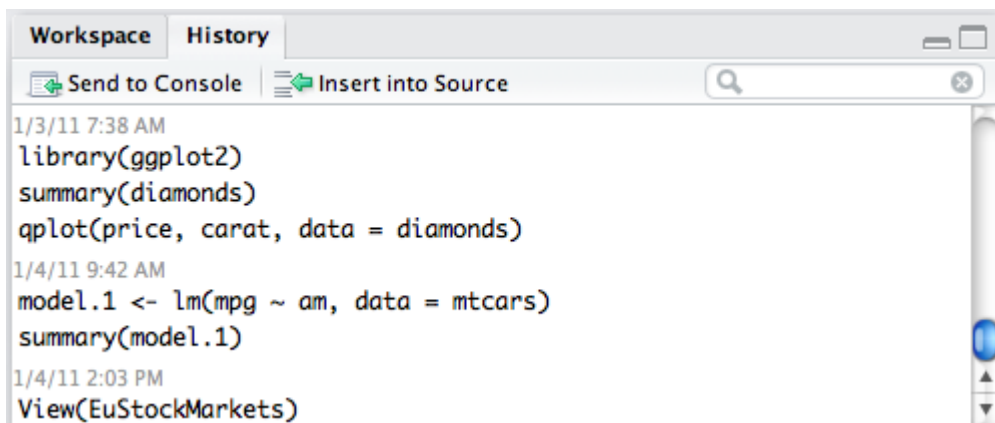
- Version control system — 프로젝트에 사용할 버전관리시스템을 정할 수 있다.

6. Command History사용하기

RStudio는 콘솔에 입력한 모든 명령을 데이터베이스에 저장한다. 이 데이터베이스는 [History]창에서 찾아 볼 수 있다.

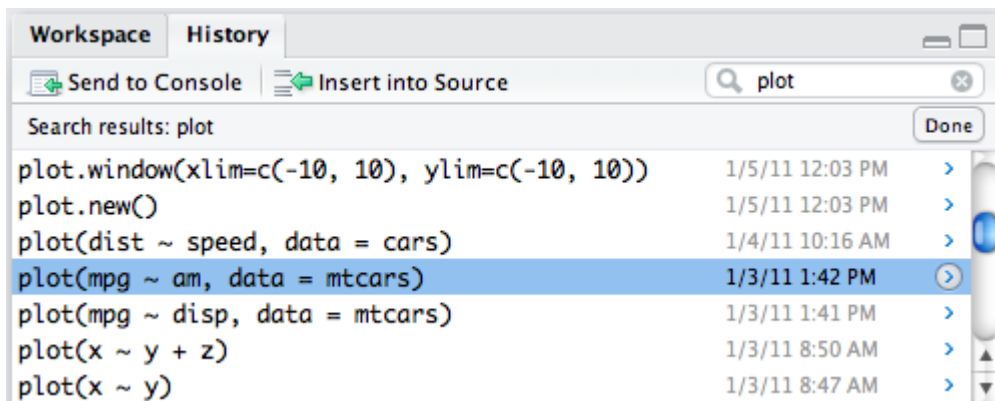
6.1명령내역 살펴보기

RStudio콘솔에 입력한 명령은 [History]탭에서 살펴 볼 수 있다. 명령은 순차적으로 표시되며 일정 시간 단위로 묶어 표시된다.



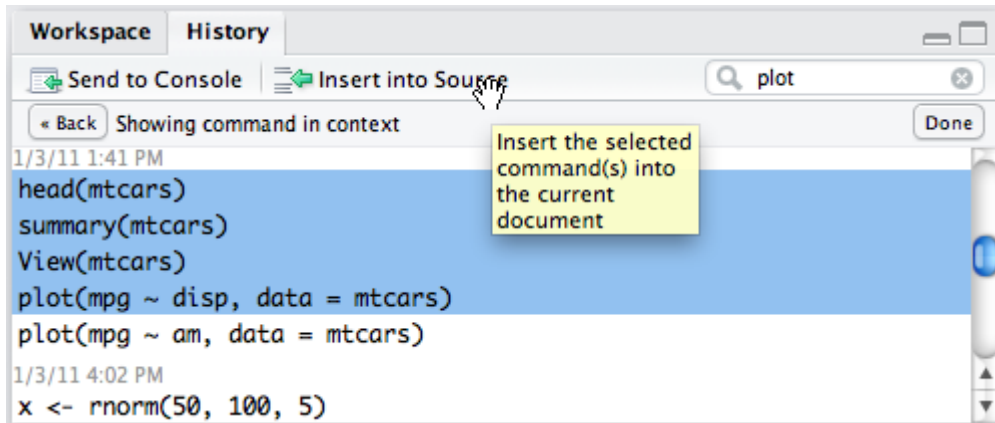
6.2 명령내역 검색

[History]탭의 오른쪽 상단에 위치한 검색상자를 사용하여 이전에 사용한 명령을 검색할 수 있다.



6.3 명령 재활용하기

[History]창에서 선택한 명령은 두 가지 방법으로 사용할 수 있다.



- Send to Console— 선택한 명령을 콘솔로 내보낸다. [Enter]키를 누르면 비로소 명령이 실행된다.
- Insert into Source— 선택한 명령을 현재 편집중인 소스에 추가한다. 현재 편집하고 있는 문서가 없다면 새로운 문서를 만들어 추가한다.

7. 작업 디렉토리와 워크스페이스(Working Directories 와 Workspaces)

R과 함께 설치되는 R GUI처럼 RStudio 는 사용자의 홈 디렉토리를 글로벌 워킹디렉토리로 사용한다. RStudio가 시작될 때 다음과 같은 일이 벌어진다

- 기본 작업디렉토리에서 .Rprofile을 실행한다.
 - 기본 작업디렉토리에서 .RData파일을 읽어서 워크스페이스로 로드한다.
- RStudio 가 종료될 때 워크스페이스의 변경사항이 있다면, 변경사항을 현재 작업디렉토리의 .RData파일에 저장할 지 여부를 묻게 된다. 그러나 이러한 사항은 옵션에서 변경가능하다.
- 기본 작업 디렉토리 변경
 - 시작시 기본 작업 디렉토리의 .RData 로딩여부
 - .RData를 항상 저장할 지, 저장 안할지, 종료시 저장여부 묻기 등을 정할 수 있다.

7.1 작업 디렉토리 변경

현재의 작업 디렉토리는 RStudio의 콘솔제목 표시줄에 표시된다. 현재의 작업 디렉토리를 변경하는 방법은 다음과 같다.

- 'setwd' R 함수를 사용한다.
- 메인 메뉴에서 [Tools]-[Change Working Directory]를 사용한다.
- [Files]작업창의 도구모음에서 [More]-[Set As Working Directory]를 사용한다

작업 디렉토리를 변경하는 경우 다음과 같은 부작용을 염두에 두어야 한다.

- 코드에서 상대적인 파일참조를 해둔 경우 실행시 파일을 참조하지 못하는 경우가 생긴다.
- 종료시 .RData의 저장위치가 새로운 디렉토리로 변경된다.

7.2 다른 작업 디렉토리에서 시작하기

7.3 워크스페이스 로드와 저장

R을 사용하는 동안 워크스페이스를 불러오거나 저장하고 싶다면 다음의 명령을 사용하여 현재의 작업 디렉토리에 .RData파일로 저장하거나 읽어 올 수 있다.

```
>save.image()
```

```
>load(".RData")
```

load 함수는 현재 워크스페이스내 개체를 완전히 바꾸는 게 아니라 덧붙이거나 겹쳐 쓴다. 따라서 워크스페이스내 현재 모든 개체를 정리하고 로드하고 싶다면 rm함수를 사용하여 다음과 같이 정리할 수 있다.

```
>rm(list=ls())
```

R의 시작과 종료시 워크스페이스 로딩과 저장이 이루어지기 때문에 대부분의 경우 위의 명령은 필요치 않다. 그러나 작업 디렉토리를 변경하는 경우 변경할 디렉토리내 워크스페이스와의 동기화를 위해 필요할 수 있다.

7.4 .Rhistory 다루기

RStudio는 .Rhistory

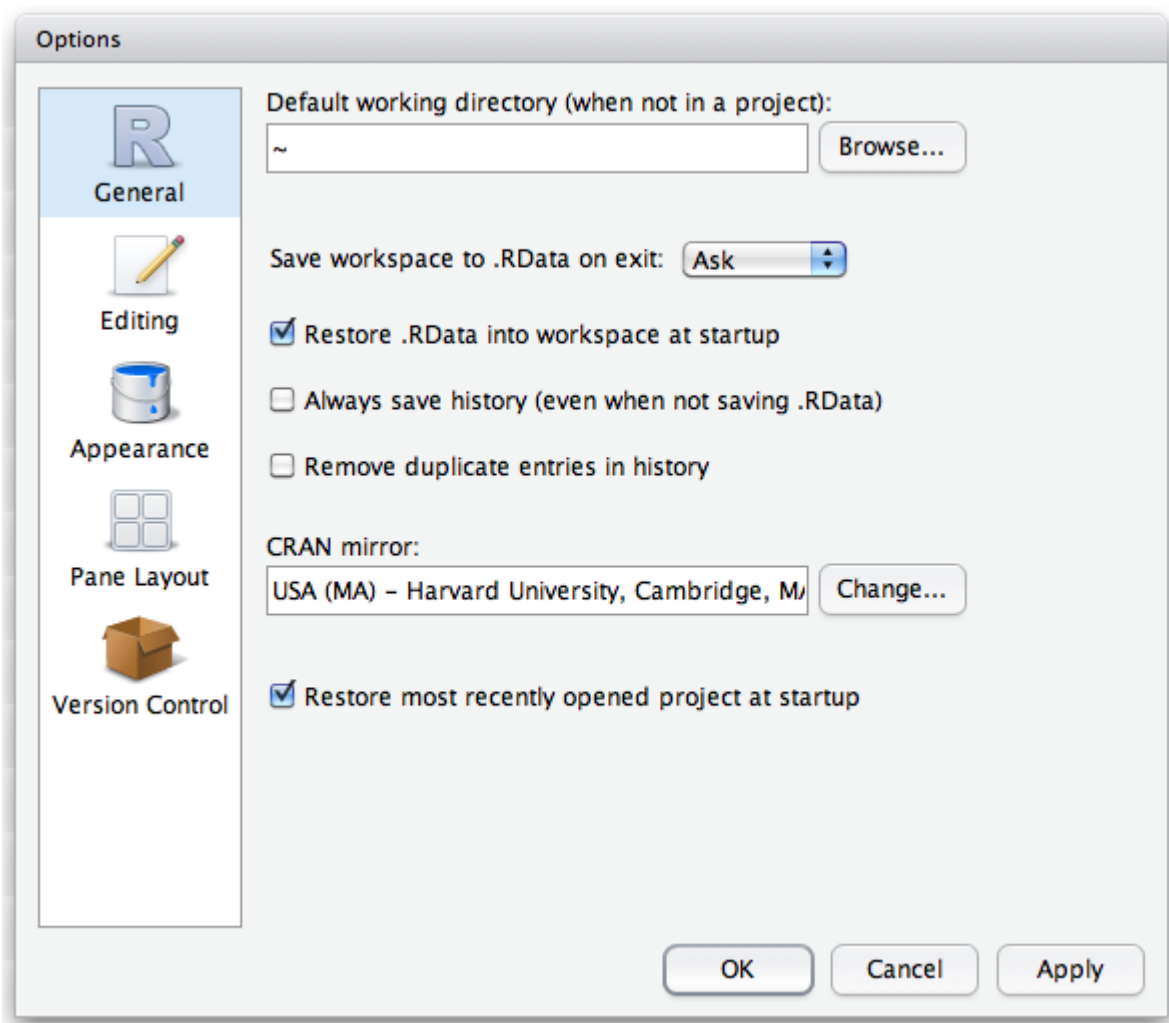
8. 입맛에 맞게 RStudio 조정하기

RStudio 옵션을 변경하려면 메인메뉴 [Tools]-[Options]를 사용한다. 여기에는 다음과 같은 부류의 선택 사항이 있다.

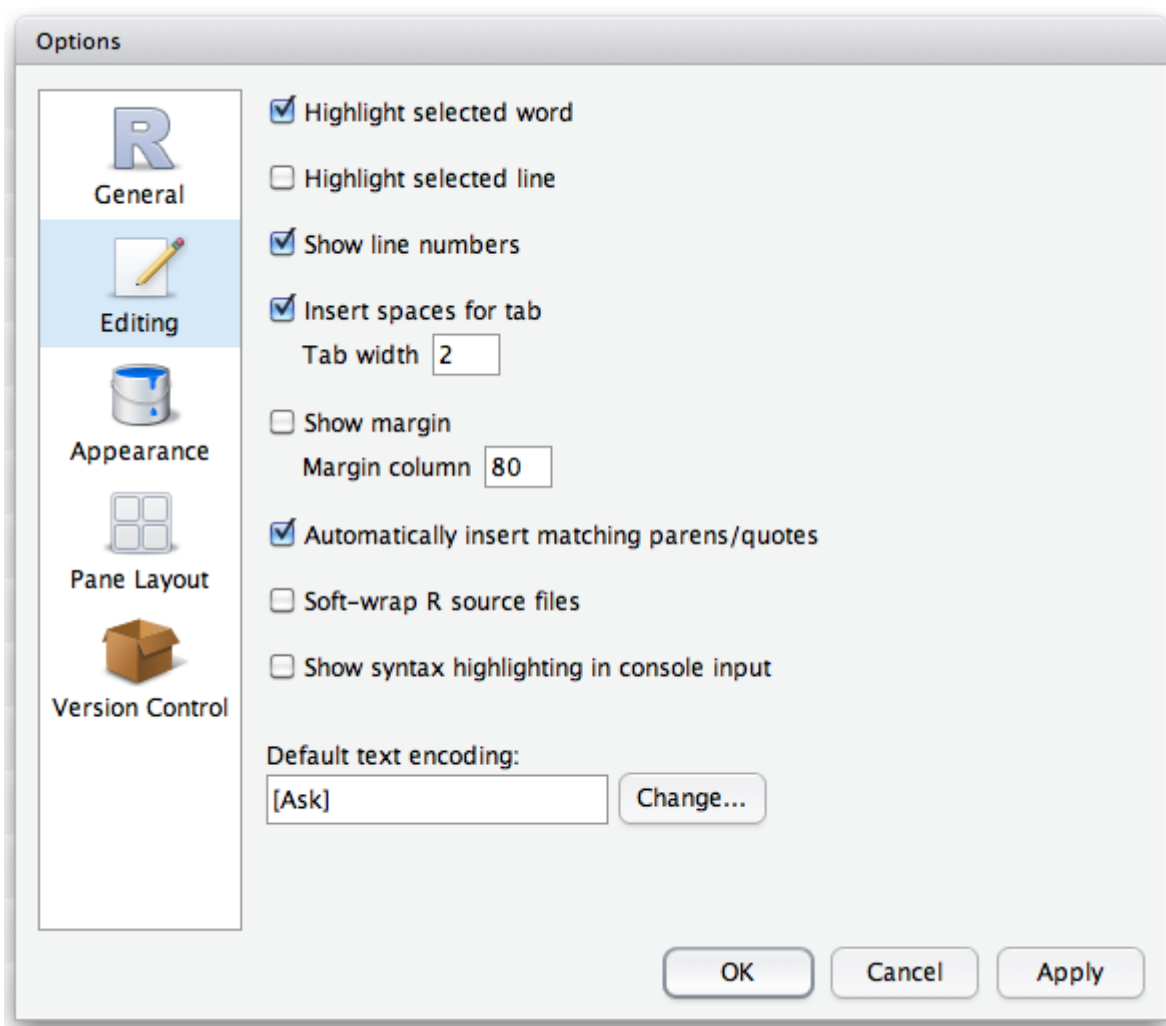
- General R Options — CRAN 미러사이트, working directory, workspace 그리고 history .
- Source Code Editing — 행 번호 표시, 선택한 단어/행 강조, soft-wrapping for R files, 자동괄호, 오른쪽 여백표시, 콘솔에서 구문강조, 탭 크기, 텍스트 인코딩
- Appearance and Themes — 글꼴크기, 콘솔과 에디터 테마
- Pane Layout —콘솔, 편집기, tab panes; 각 Pane내 항목
- Version Control — 버전관리시스템인 Git and Svn위치, SSH RSA keys 생성/보기.

□

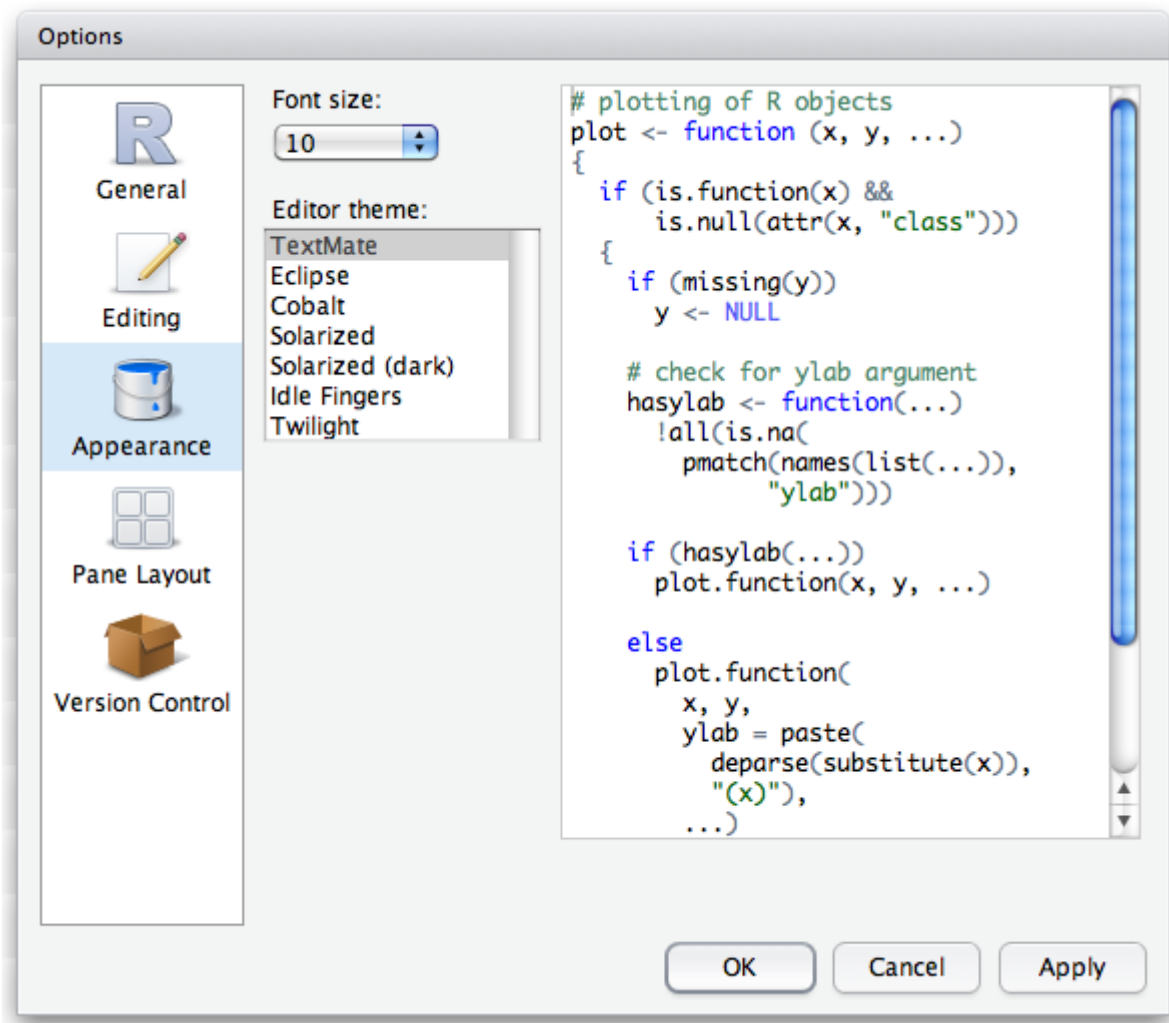
8.1 General R Options



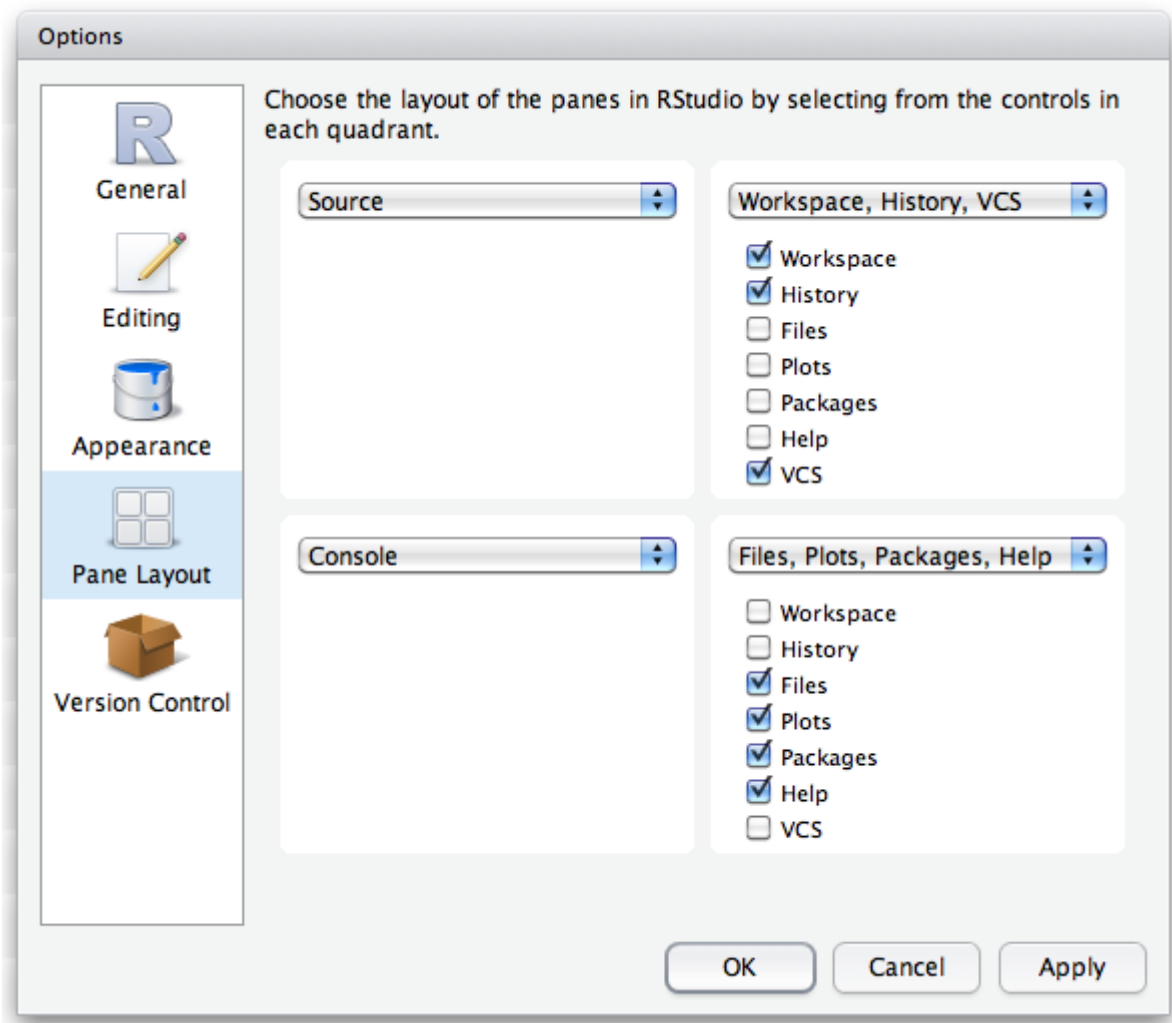
8.2 Editing



8.3 Apperance



8.4 Pane Layout



8.5 Version Control

Options



General



Editing



Appearance



Pane Layout



Version Control

☒ Enable version control interface for RStudio projects

Git executable:

/usr/local/git/bin/git

Browse...

SVN executable:

/usr/bin/svn

Browse...

SSH RSA Key:

[View public key](#)

~/.ssh/id_rsa

Create RSA Key...

? [Using Version Control with RStudio](#)

OK

Cancel

Apply

9. 단축 키

콘솔 단축키

편집기 단축키

편집시(콘솔과 편집기 모두)

코드완성(콘솔과 편집기 모두)

보기

차트그리기

Git/Svn

10. RStudio 콘솔 언어 변경하기

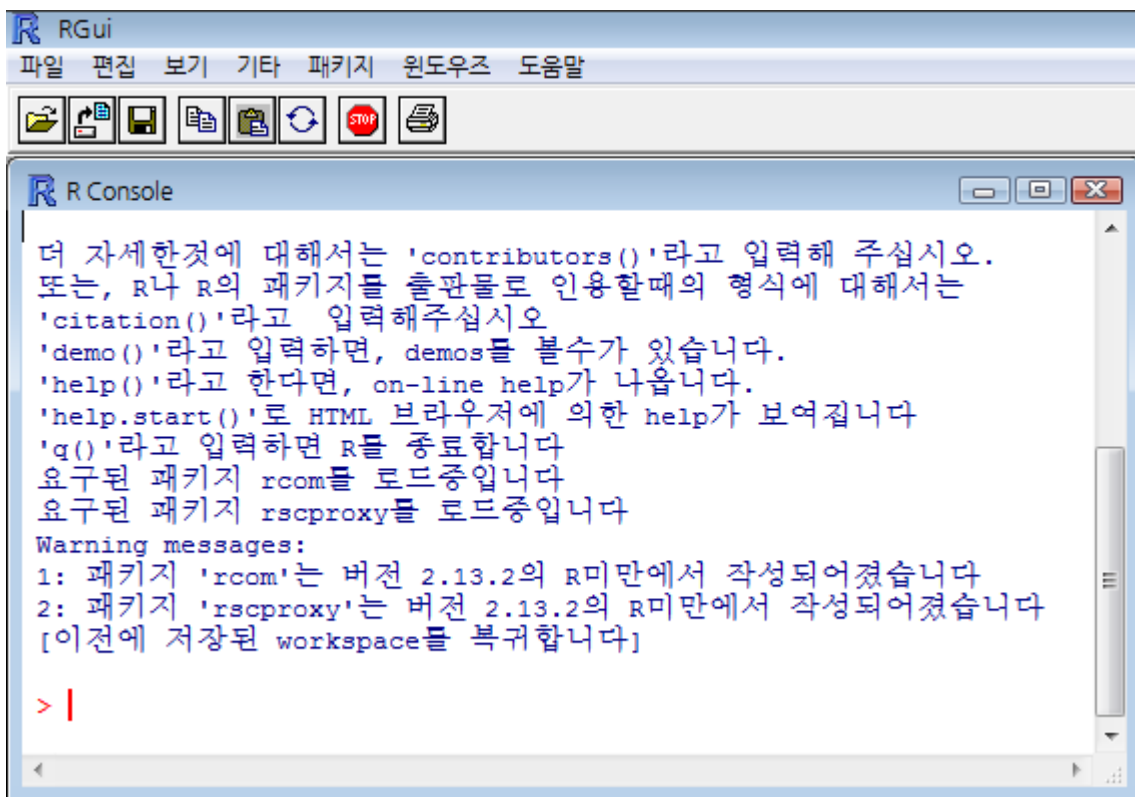
최근에 많은 외산프로그램들이 한글을 잘 지원하지만, 너무 잘 지원해서 문제인 경우도 있다. RStudio 콘솔에서 아래와 같은 명령을 사용하다가 한글화된 에러 메시지를 받는 경우가 있다.

```
> apply(1:5,1,log)
```

이하에 에러 `apply(1:5, 1, log)` :

`dim(x)`은 positive 길이를 가지지 않으면 되지 않습니다

난감한 이유는 우선 한글이지만 무슨 말인지 모르겠다는 점이다. 그래서 검색을 해보려니 R사용자가 국내에 많지 않아 검색결과가 적을 수 있고 따라서 해결책을 찾지 못할 수 있다는 점이다. 차라리 영어로 나오면 검색결과가 많아 쉽게 해결할 것이다.



그러고 보니 R도 그렇고 RStudio도 모두 콘솔의 언어는 한글이다. 심지어 RGui의 메뉴도 어설픈 한글이라 허접하다는 느낌을 지울 수 없다. RGui는 잘 사용하지 않으니 넘어가고, 대신 자주 사용하는 RStudio만이라도 한글메시지를 영문으로 바꾸는 방법이 없을 까?

RStudio.org에선 이것은 R의 언어설정에 따른다고 한다. 방법은 RGui의 [편집]-[GUI설정]을 클릭하면 언어설정을 변경할 수 있다.

RGui 구성 편집기

Single or multiple ☒ MDI ☐ SDI ☒ MDI toolbar ☐ MDI statusbar

Pager style ☒ multiple windows ☐ single window Language for menus and messages

Font ☒ TrueType only size style

Console rows columns Initial left top
☒ set options(width) on resize? buffer chars lines
☒ buffer console by default? Cursor blink

Pager rows columns

Graphics windows: initial left top


Console and Pager Colours

| | | |
|------------|--------|-------------|
| background | wheat2 | Sample text |
| normaltext | wheat3 | |
| usertext | wheat4 | |
| pagerbg | white | |
| | | |

Apply Save... Load... OK Cancel

오른쪽 상단의 'Language for menus and messages' 옆에 빈 텍스트 상자에 입력하면 된다. 영문은 'en', 한글은 'ko'이다. 영어로 변경할 거라서 'en'을 입력하고 [Save]를 클릭하여 환경설정파일인 'RConsole'을 저장한다(내 경우엔 '내 문서' 폴더에 저장했음) 저장하지 않고 [OK]하면 도루묵이다.

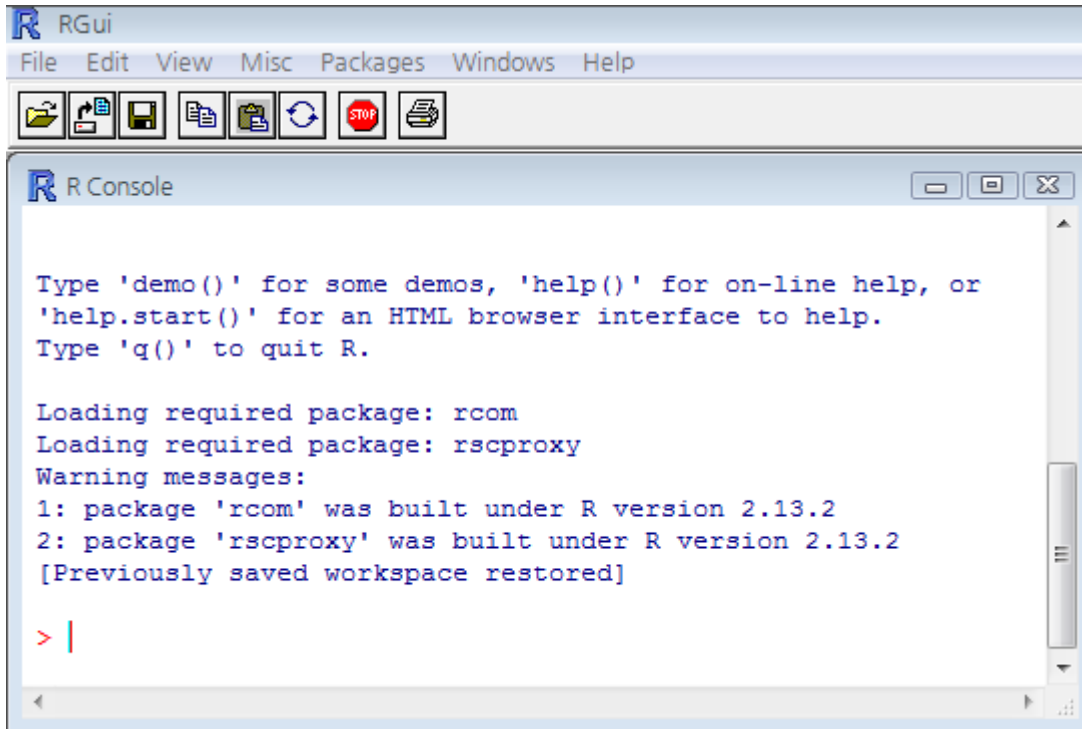
Information

 실행 중인 콘솔에서 메뉴 언어를 바꿀 수 없습니다.

메뉴에 적용하려면 설정을 저장하고 Rgui를 다시 시작해주시오.

확인

RGui를 다시 실행하면 영어 메뉴와 메시지를 볼 수 있다.



이제 다음과 같이 영어로 된 메시지를 RStudio에서 볼 수 있다.

